Deep Learning Enhanced Time-step Control in Pseudo Transient Analysis for Efficient Nonlinear DC Simulation

Xiaru Zha[1], Haojie Pei[1], Dan Niu[2], Xiao Wu[3*] and Zhou Jin[1*]
1.Super Scientific Software Laboratory, China University of Petroleum-Beijing, Beijing, China
2.School of Automation, Southeast University, Nanjing, China
3.Huada Empyrean Software Co. Ltd, Beijing, China
Email: 2019011712@student.cup.edu.cn, haojiepei@student.cup.edu.cn, 101011786@seu.edu.cn, wuxiao@mail.empyrean.com.cn, jinzhou@cup.edu.cn

***Abstract*—With the development of VLSI, circuit simulation plays an important role in circuit design, and DC analysis, as the basis of circuit behavior analysis, is the foundation for nonlinear electronic circuit simulation. Pseudo transient analysis(PTA) methods have gained great success among various continuation algorithms. However, PTA tends to be computationally intensive without proper time-step control method. In this paper, we harness the latest advancing in deep learning to resolve this issue. Particularly, a coarse and fine grained hybrid sampling strategy is used to search the optimal time-step, which resolves the problem that the optimal time-step is no precise definition in PTA theory. Afterwards, the long short-term memory(LSTM) algorithm can be utilized to learn an optimal time-step control method through feature selection and two-stage data preprocessing strategy accelerating DC analysis ultimately. Experimental results demonstrate a significant speedup of up to 41.98X.**

***Index Terms*—Circuit Simulation, Nonlinear DC Analysis, Pseudo Transient Analysis, Time-Step Control, Deep Learning**

1. Introduction

With the development of semiconductor technology, the integration and complexity of integrated circuits show an exponential growth trend and DC analysis, as the basis of circuit simulation, is also required to solve very large scale and strong nonlinear algebraic system constructed by modified nodal analysis(MNA) [1] efficiently. As is well-known, Newton-Raphson(NR) method is the most commonly used method to solve nonlinear algebraic equations because of its quadratic convergence. However, when NR method solves very large scale and strong nonlinear algebraic system, NR does not converge frequently [2]. Therefore, the continuation method of DC analysis is widely studied, including PTA [3], Gmin stepping [4], source stepping [5], homotopy [6], etc. Unfortunately, the convergence of Gmin stepping and source stepping are often inferior in strong nonlinear DC analysis. Similarly, although homotopy can guarantee global convergence, its realization is highly dependent on device model. Therefore, among the continuation methods, PTA is expected to solve the very large scale and strong nonlinear DC analysis, and PTA has been proved to be the most promising method because it is easy to implement and has no discontinuity issue.

PTA is a method that transforms nonlinear algebraic systems, which are difficult to solve directly, into ordinary differential systems with initial value problems by inserting pseudo-elements. Once the PTA solver forms the ordinary differential system, it is solved iteratively to the steady state using numerical integration methods based on time-step control. However, the efficiency of PTA is subject to the time-step control method, which determines the discrete time points that need to be solved, including the time-consuming and resource-consuming NR iterations. Some time-step control methods based on simple formulas have been proposed to accelerate PTA, as described in previous studies [7], [8]. However, these methods quickly become inadequate for simulating very large and strongly nonlinear systems. Therefore, there is a pressing need for a more effective time-step control method.

Fortunately, the rise of deep learning has enabled the solution of many complex problems, including computer vision [9] and natural language processing [10], presenting an opportunity for the development of efficient time-step control methods. However, such methods must overcome several challenges. (1) While different circuit types have varying time-step requirements, common process variables can be used as features, based on expert experience, to identify these requirements. (2) An optimal time-step sampling strategy needs to be explored since PTA theory lacks a precise definition for the optimal time-step. (3) Timing information is crucial for PTA time-step control, so any proposed algorithm must have the ability to process this information. This paper proposes an optimal time-step control method enhanced by deep learning, which solves the first challenge. It also contains a hybrid sampling strategy, combining coarse and fine-grained approaches, to solve the second challenge. Lastly, it uses long short-term memory (LSTM) to address the third challenge. By solving these challenges, our contributions will be discussed as follows.

(1) The optimal time-step is approximated by coarse and fine grained hybrid sampling strategy, which solves the problem that the optimal time-step cannot be defined theoretically.

(2) The time-step control method enhanced by deep

learning and based on feature selection and two-stage data preprocessing strategy has better generalization and simulation efficiency.

(3) The proposed method has been implemented in an out-of-the-box SPICE-like simulator and is verified by benchmark circuits. Significant acceleration is achieved, i.e., a maximum 41.98X speedup is demonstrated on practical circuits.

2. Preliminaries

2.1. Pseudo Transient Analysis

PTA [7], [11], [12] transforms original hard-to-solve nonlinear algebraic systems

$$\boldsymbol{F}(\boldsymbol{x}) = 0 \tag{1}$$

(where $\boldsymbol{F}(\cdot): R^m \to R^m$, $\boldsymbol{x} = (\boldsymbol{v}, \boldsymbol{i})^T \in R^m$, $m = N+M$, variable vector $\boldsymbol{v} \in R^N$ denotes node voltage, and vector $\boldsymbol{i} \in R^M$ represents internal branch current) into ordinary differential systems

$$\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{D} * \dot{\boldsymbol{x}}(t) = 0 \tag{2}$$

(where $\dot{\boldsymbol{x}}(t) = (\dot{\boldsymbol{v}}(t), \dot{\boldsymbol{i}}(t))$, and $\boldsymbol{D}$ represents for the incidence matrix of inserted pseudo-elements) with an initial value problem by inserting specific pseudo-elements such as capacitors and inductors into the circuit.

Implicit numerical integration algorithms, e.g.(3), are used to discretize the time-domain, and the steady state is obtained through iterative difference approximation of the differential term.

$$\dot{\boldsymbol{x}}(t)|_{t=t_{n+1}} = (\boldsymbol{x}_{n+1} - \boldsymbol{x}_n)/h_{n+1} \tag{3}$$

The process of choosing the appropriate h value in Eq.(3) for each iteration in PTA is referred to as the time-step control method.

2.2. Time-step Control Method

There are two time-step control methods based on simple formulas. The conventional PTA methods use a simple iteration counting method [7] to determine the time-step size. This method employs time-step control through two options (IMAX and IMIN). The number of NR iterations at each time-point is compared with these options to determine the next time-step. The advantage of this method is that the time-step can be increased quickly and easily. However, it is challenging to select appropriate parameters, including IMAX, IMIN, initial time-step, and time-step growth rate, for different circuits.

The adaptive time-step control method was proposed in [8] based on the Switched Evolution/Relaxation (SER) method, which uses the following equation to control the time-step.

$$\begin{aligned} h_{n+1} &= E(h_n, Nitr_n, \mathbf{x}, \boldsymbol{F}(\boldsymbol{x})) \\ &= h_n \cdot MAX(1, \delta \cdot \gamma \cdot \|\boldsymbol{F}(\boldsymbol{x})\|) \end{aligned} \tag{4}$$

This heuristic method employs domain experience and has demonstrated great potential in speeding up intelligent time-step control. However, it still does not guarantee that the time-step is always as large as possible.

2.3. Long Short-Term Memory

Long short-term memory (LSTM) is a type of recursive neural network (RNN) [13] that can analyze time series and overcome the vanishing gradient issue. It is commonly used in sequential tasks that involve time-dependent data, such as speech recognition and machine translation. Unlike traditional RNNs, which suffer from the problem of long-term dependence, LSTM was designed to address this issue from the outset. This allows LSTM to effectively convey and express information from long time series without forgetting useful information from the distant past. The LSTM architecture is composed of three gate structures, namely the forget gate, input gate, and output gate. These gates allow for the retention and management of temporal information. The forget gate, as shown in Eq.(5), serves the purpose of deciding which information should be forgotten.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

The input gate, as expressed in Eq.(6), is responsible for determining which new information should be stored and which existing information should be updated.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C}_t &= tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= i_t \cdot \widetilde{C}_t \end{aligned} \tag{6}$$

The output gate, as shown in Eq.(7), controls which information should be propagated to the next cell.

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \cdot tanh(C_t) \end{aligned} \tag{7}$$

3. Proposed methods

3.1. Overview

The time-step control for PTA method is not determined by accuracy considerations. Instead, the time-step is made as large as possible, consistent only with the convergence of the NR iteration [8]. Based on these characteristics of time-step control in PTA, we introduce the coarse and fine grained hybrid sampling strategy to find the optimal time-step that allows for the largest time-step while ensuring the convergence of the NR iteration.

There are two kinds of optimal time-step. In one case, when the sampling strategy is coarse-grained, the time-step is increased according to the conventional time-step control. When the previous NR iteration converges, but the current NR iteration does not, the fine-grained search is triggered. In this case, the time-step decreases with a certain granularity from large to small until a convergent time-step is found and marked as the optimal time-step. In the other case, the current NR iteration converges and the time-step has reached the maximum time-step. And the maximum time-step is marked as the optimal time-step in this case.

After obtaining the optimal time-step as described above, the proposed method involves mapping time-step control as a regression prediction problem. This is achieved by fitting

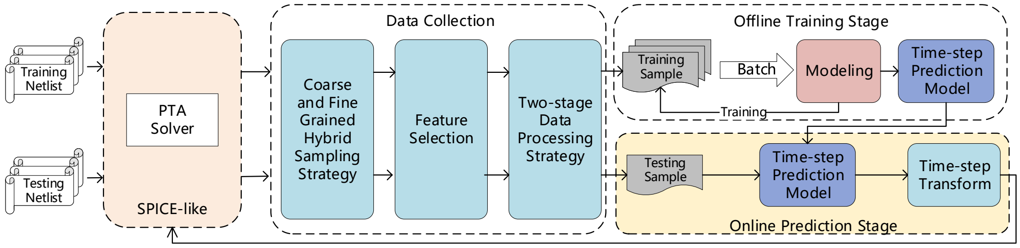


Figure 1. Entire flow of proposed method

an optimal time-step control function between time-step h and selected features $\boldsymbol{s}$ on training set.

$$h = f(\boldsymbol{s}, \theta) \tag{8}$$

, the parameters θ need to be learned during training. Then, we use LSTM on the training set to find parameters θ^* on training set, which makes model $f(\boldsymbol{s}, \theta^*)$ be closed to the actual optimal time-step control function $f(\boldsymbol{s}, \theta)$. As a result, the differential system reaches the steady state rapidly, as demonstrated in Eq.(9).

$$NR_{iters} = \lim_{n \to N_{stop}} \min_{\theta^*} \boldsymbol{F}(\boldsymbol{x}_n) + \boldsymbol{D} \cdot \frac{\boldsymbol{x}_n - \boldsymbol{x}_{n-1}}{f_n(\boldsymbol{s}, \theta^*)} = 0 \tag{9}$$

The entire flow of the proposed method is illustrated in Figure1. During the PTA iteration, samples are collected for offline model training. The trained model is then used for online prediction during the subsequent PTA iterations. Note that PTA itself is an ordinary differential system over time, thus LSTM is quite suitable for this work due to its superiority in processing time series information.

3.2. Coarse and Fine Grained Hybrid Sampling

As mentioned previously, the first task to be addressed is the sampling of the optimal time-step. However, there is no precise definition of what constitutes an optimal time-step in PTA. The PTA method comprises two layers of iterations, an outer PTA iteration, and an inner NR iteration. The objective of the sampling process is to find the optimal time-step for each PTA iteration. In order to achieve efficiency, a larger time-step that ensures NR convergence results in fewer discrete time points that need to be computed. Conversely, non-convergence of NR leads to rollback, necessitating additional computation. Therefore, the optimal time-step is approximated by the largest possible time-step that guarantees NR convergence. To obtain the optimal time-step, we introduce the coarse and fine-grained hybrid sampling method, as shown in Fig.2.

The reason why the above flow uses the combination of coarse and fine granularity to search the time-step is that the result of PTA iteration is too radical to obtain the optimal time-step precisely under the coarse grained decrease based on the conventional method. At the same time, in order not to affect the sampling efficiency, we add a fine-grained search, which can select automatically the proper granularity according to the dimension of time-step to reduce the time-step, so that the optimal time-step under this granularity can be obtained accurately and quickly.

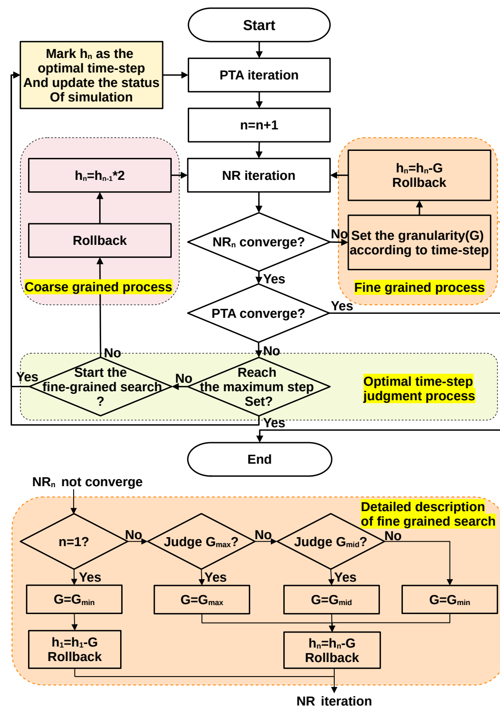


Figure 2. Coarse and fine grained hybrid sampling strategy

The presented sampling strategy has been shown to provide a high-quality dataset for implementing deep learning enhanced optimal time-step control, as demonstrated by the speedup results of several simulated circuits listed in Table1.

TABLE 1. Verify the validity of the coarse and fine grained hybrid sampling strategy on several circuits

circuit	conventional	proposed sampling	speedup
fadd32	1968	121	16.26
ab_opamp	2417	213	11.35
ab_integ	4540	159	28.55
schmitfast	5681	68	83.54
THM5	5331	80	66.64

3.3. Feature Selection

Appropriate feature selection not only reduces the computational burden of model training but also improves the accuracy of prediction. Furthermore, after the feature selec-

TABLE 2. FEATURES AND DESCRIPTIONS

Features	Brief Descriptions	Data Type
NRs_{n-1}	Evaluate the difficulties of NR convergence at previous optimal time-step	Scalar
$Res_{n-5:n-1}$	Evaluate whether equation is close to final solution at five discrete time points respectively	Vector
$Time-step_{n-1}$	The previous optimal time-step	Scalar
$Vol_{n-5:n-1,1:10}$	The ten voltage solution curves in descending order of fluctuation	Matrix

tion process is complete, we need to simulate all training netlists to generate the complete dataset.

The time-step control in PTA is not restricted to the circuit but is dependent on the change trend of the process variable in the simulation. Hence, we select process variables from the simulation as features instead of relying on circuit-specific features like circuit type. This approach allows us to build a sample set that is not limited to a particular circuit type and can be sampled from the simulation process of all circuits to obtain as many samples as possible. In addition, according to PTA and expert experience, the symbolic representation, brief description, and data types of the selected features are shown in Table2. It is important to note that we use the features from five consecutive time points to predict the sixth time-step. This is because too few time points are insufficient to adequately capture the voltage fluctuation of nodes. Additionally, in order to address the issue of inconsistent numbers of features arising from different numbers of nodes across circuits, we uniformly select the ten solution curves with the largest fluctuation for each circuit. The detail description of important features are given following.

$\boldsymbol{NRs}_{n-1}$ represents the difficulties of NR convergence at previous optimal time-step. A smaller value of it indicates that NR converges more easily, allowing for a larger next time-step.

$\boldsymbol{Res}_{n-5:n-1}$ represents the distance of PTA convergence. A larger time-step can also ensure the convergence of NR when the residual value enters the PTA convergence stage.

$\boldsymbol{Time-step}_{n-1}$ represents time-step at previous optimal time-step, which is basement for next time-step.

$\boldsymbol{Vol}_{n-5:n-1,1:10}$ represents ten voltage solution curves with the largest fluctuation, which are related to the time-step used in the simulation. Typically, a smaller time-step is needed for simulations with more dramatic voltage fluctuations.

3.4. Two-stage Data Preprocessing Strategy

Inside data preprocessing strategy. Based on the data types in Table2, it is evident that different features have varying data types. However, for the model to function properly, it requires input features with uniform data types, specifically one-dimensional row vectors. Consequently, there is a need for unification of residuals and voltages. Firstly, for the voltage with matrix type, in order to describe the fluctuation of each voltage solution curve, the variance is adopted to normalize each voltage solution curve and a one-dimensional row vector of size 10 can be obtained. Secondly, for residuals with a one-dimensional column vector type, we utilized the standard deviation to normalize the residuals and obtain a scalar value, which describes the convergence distance of the current equations. Then, we concatenated independent features based on the column direction to form a one-dimensional row vector. It is worth noting that the range of time-step may vary by several orders of magnitude, which can increase the difficulty of model learning. To simplify the learning process, we converted time-step prediction to a prediction based on the multiples of the previous step. In Table3, the effectiveness of the prediction based on the time-step multiples of the previous step is verified by the speedup for models with two different labels on several circuits.

TABLE 3. VERIFY THE EFFECTIVENESS OF CONVERTING TIME-STEP PREDICTION INTO MULTIPLES BASED PREDICTION ON SEVERAL CIRCUITS

circuit	time-step(#iters)	ours(#iters)	speedup
g1310	121	56	2.16
hussamp	1365	240	5.69
D2	545	56	9.73
DCOSC	654	188	3.48
UA709	2160	711	3.04
UA733	1073	133	8.07

Outside data preprocessing strategy. After performing internal data processing, we concatenated all processed one-dimensional row vectors into a large matrix to form the training set. As we are aware, using a training set with different value ranges and dimensions for each column feature can significantly increase the training time and even result in non-convergence of the model. Hence, to ensure numerical consistency, we utilized maximum and minimum normalization for each column of the training set.

3.5. Modeling and Training

As mentioned earlier, the LSTM deep learning model is particularly effective in processing timing information and avoiding the gradient disappearance problem commonly associated with traditional RNN. In our work, we leveraged the PyTorch machine learning library to construct and train our LSTM model using all available data. Specifically, we designed the LSTM network structure to include four hidden layers, each with 120 cells and ReLU activation functions. For optimization, we utilized a batch size of 32, learning rate of 0.0005, Adam optimizer, and mean square error loss function. Finally, we used Alg.1 to process the data and train our model. Upon completion of the training process, we obtained a reliable model capable of accurately predicting the time-step.

Algorithm 1 Deep learning enhanced time-step control method for PTA

Input: Training netlists ξ
Output: Time-step predictor $f(s, \theta^*)$

Coarse and Fine Grained Hybrid Sampling Strategy
Construct nonlinear equation $F(x)$ by ξ
for PTA is not converge **do**
  Execute $NR_{solver}F(x)$
  Find the maximum time-step that ensures NR convergence and mark as the optimal time-step
end for
Obtain optimal time-step set $H_{1:n}$
Feature Selection
Select features $S_{1:n,1:k}$
Two-stage data preprocessing strategy
Execute $S'_{1:n,1:k}, \hat{H}_{1:n} = Inside(S_{1:n,1:k}, H_{1:n})$
Execute $\hat{S}_{1:n,1:k} = Outside(S'_{1:n,1:k})$
Modeling and Training
Construct LSTM model with trainable parameters θ
for i to n / $Batch$ **do**
  Loss(LSTM($\hat{S}_{1:k,i:i+Batch}$, $\hat{H}_{1,i:i+Bactch}$, θ))
  Update $\theta \leftarrow \theta$
  Update $i \leftarrow i + Batch + 1$
end for
$f(\cdot) \leftarrow LSTM(\cdot)$

4. Experiment results

4.1. Experimental Setup

The deep learning model of the proposed method is trained using python and the torch framework, implemented in the WSPICE simulator based on SPICE3f5. A total of 745 samples from 5 circuits are used to train the model and obtain the final version. We compared the efficiency and robustness of the proposed method with those of the conventional PTA algorithm and adaptive PTA algorithm by examining the total number of NR iterations used during simulation. All circuits used in the experiments were selected from benchmark [14] as well as from our laboratory.

4.2. Acceleration Simulation Efficiency

To demonstrate the effectiveness of the proposed method in improving simulation efficiency, we compare it with the adaptive method and conventional method. For each time-step control method, pseudo-elements are inserted into the transistors between each node and ground using the diagonal embedding position [15], to improve the convergence effect. We evaluate the efficiency in terms of the number of NR iterations on 11 testing circuits, and the results are presented in Table4. The proposed method shows a significant improvement over the conventional method, achieving up to 41.98X speedup, and over the adaptive method, achieving up to 41.92X speedup, in damped pseudo-transient analysis(DPTA). DPTA, a variation of PTA, solves oscillation problems by artificially increasing the damping effect in the numerical integration algorithm. Furthermore, we demonstrate the use of circuit features such as nodes, bjt, mos, etc, indicating that our algorithm has superior generalization capabilities - a crucial metric in deep learning.

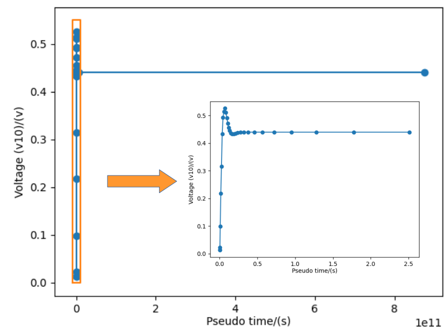


Figure 3. The node voltage waveform of "THM5" circuit by using proposed method

Furthermore, the practical circuit "THM5", which contains 1 voltage sources and 9 bjt transistors, is selected by us for more detailed analysis. Generally the time-step control can be categorized into two distinct phases in PTA: the search phase and the convergence phase. Fig.3 shows the voltage curve of node 10, and the solution of each pseudo-step is close to the final solution in the convergence phase. It can be clearly seen that the proposed algorithm can provide a larger time-step, thus reducing the discrete time points in the convergence stage. Similarly, the subfigure of Fig.3 shows the voltage curve of node 10 in the search stage. As observed, in the search phase there are the time-step is small and numerous time points. Because the voltage changes from zero to full value in the search phase, which leads to a relatively large fluctuation of the voltage curve. And a small time-step is able to ensure the convergence of NR. In addition, the proposed algorithm will not change the continuity of the voltage curve or cause the oscillation of the voltage curve, so it has better practicability.

4.3. Improvement Simulation Convergence

It's worth noting that the proposed method demonstrates the ability to solve non-convergence(N/A) issues for certain DPTA cases, as shown in Table5. This is especially valuable for PTA based on SPICE, as non-convergence problems can be very challenging for simulators to handle, and their root causes are often difficult to identify accurately. The proposed algorithm significantly enhances the robustness of DPTA, which is an important practical consideration in real-world applications.

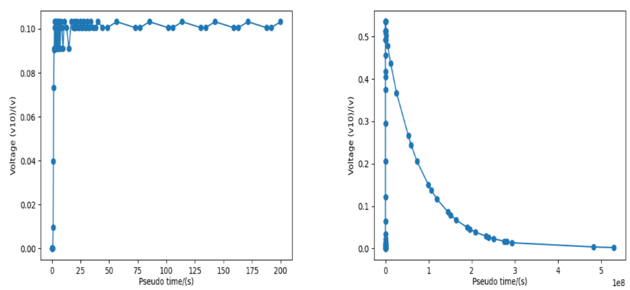
A large circuit named "voter" is used to illustrate the advantages of the proposed algorithm. The "voter" circuit consists of 4,243 MOS transistors and 23 voltage sources. In Fig.4(a), a portion of the voltage curve of node 10 in the "voter" circuit is shown under the conventional time-step control method. It is evident that the voltage curve exhibits oscillations and does not converge, which is a limitation of the conventional time-step control method. In contrast, the proposed algorithm overcomes this limitation and prevents oscillations, as depicted in Fig.4(b), thereby improving the convergence of DPTA.

TABLE 4. CIRCUIT CHARACTERISTICS AND SIMULATION EFFICIENCY FOR DPTA

circuit	nodes	eqn	bjt	mos2	mos3	c	r	v	number of iters			speedup	
									conventional	adaptive	ours	vs. conventional	vs. adaptive
nagle	26	54	23	0	0	1	11	5	2093	1948	672	3.11	2.90
ab_ac	25	28	0	31	0	22	1	3	3961	3947	265	14.95	14.89
ab_integ	28	32	0	31	0	24	3	4	4540	4406	402	11.29	10.96
ab_opamp	28	31	0	31	0	24	4	3	2417	2536	430	5.62	5.90
e1480	145	204	0	28	0	17	130	3	5553	5514	369	15.05	14.94
mosrect	6	10	0	4	0	0	2	2	838	826	84	9.98	9.83
schmitfast	5	19	0	6	0	0	0	2	5681	5691	176	32.28	32.34
slowlatch	12	37	0	0	14	0	1	5	9382	9353	264	35.54	35.43
fadd32	161	178	0	288	0	25	0	17	1968	1859	284	6.93	6.55
TADEGLOW6TR	18	18	0	3	0	0	18	1	145	102	70	2.07	1.46
THM5	26	26	9	0	0	0	0	1	5331	5324	127	41.98	41.92

TABLE 5. IMPROVEMENT CONVERGENCE FOR DPTA ON SOME CIRCUITS

circuits	convergence		
	conventional	adaptive	ours
bjtff	N/A	N/A	479
schmitslow	N/A	N/A	468
toronto	N/A	N/A	364
add20	N/A	N/A	673
mem_plus	N/A	N/A	858
ram2k	N/A	N/A	526
voter	N/A	N/A	1261
jge	N/A	N/A	1342



(a) (b)

Figure 4. Comparison of the node voltage waveform on "voter" circuit at search phase using two time-step control methods

5. Conclusion

In this paper, we propose a novel time-step control method that uses deep learning to efficiently complete nonlinear DC analysis in DPTA. Our approach utilizes coarse and fine grained hybrid sampling strategy and two-stage data processing strategy to create a high-quality dataset for training the model. Our numerical experiments demonstrate that our method achieves a significant speedup of up to 41.98X compared to other time-step control methods. Importantly, we show that our approach can also help address non-convergence issues that can arise in some circuits due to time-step control.

Acknowledge

We deeply appreciate the invaluable comments from the reviewers. Zhou Jin and Xiao Wu are the corresponding authors of this paper. This work was supported by National Key RD Program of China (Grant No. 2022YFB4400400), the Key Program of the National Natural Science Foundation of China (Grant No.62204265, 62234010), State Key Laboratory of Computer Architecture (ICT, CAS) under Grant No. CARCHA202115.

References

[1] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on circuits and systems*, vol. 22, no. 6, pp. 504–509, 1975.

[2] L. O. Chua, "Computer-aided analysis of electronic circuits," *Algorithms and computational techniques*, 1975.

[3] F. N. Najm, *Circuit simulation*. John Wiley & Sons, 2010.

[4] K. Kundert, *The Designer's Guide to SPICE and SPECTRE®*. Springer Science & Business Media, 2006.

[5] T. Najibi, "Continuation methods as applied to circuit simulation," *IEEE Circuits and Devices Magazine*, pp. 48–49, 1989.

[6] C. Lemke, "Pathways to solutions, fixed points, and equilibria (cb garcia and wj zangwill)," *SIAM Review*, pp. 445–446, 1984.

[7] X. Wu, Z. Jin, D. Niu, and Y. Inoue, "A pta method using numerical integration algorithms with artificial damping for solving nonlinear dc circuits," *IEICE*, pp. 512–522, 2014.

[8] X. Wu, Z. Jin, D. Niu, and Y. Inoue, "An adaptive time-step control method in damped pseudo-transient analysis for solving nonlinear dc circuit equations," *IEICE*, pp. 619–628, 2017.

[9] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

[10] E. Cambria and B. White, "Jumping nlp curves: A review of natural language processing research," *IEEE Computational intelligence magazine*, vol. 9, no. 2, pp. 48–57, 2014.

[11] H. Yu, Y. Inoue, K. Sako, X. Hu, and Z. Huang, "An effective spice3 implementation of the compound element pseudo-transient algorithm," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 90, no. 10, pp. 2124–2131, 2007.

[12] C. T. Kelley and D. E. Keyes, "Convergence analysis of pseudo-transient continuation," *SIAM Journal on Numerical Analysis*, pp. 508–523, 1998.

[13] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[14] J. Barby and R. Guindi, "Circuitsim93: A circuit simulator benchmarking methodology case study," in *Sixth Annual IEEE International ASIC Conference and Exhibit*, pp. 531–535, 1993.

[15] Z. Jin, X. Wu, D. Niu, and Y. Inoue, "Effective implementation and embedding algorithms of cepta method for finding dc operating points," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 96, no. 12, pp. 2524–2532, 2013.